

Remote health-care monitoring using Personal Care Connect

M. Blount
V. M. Batra
A. N. Capella
M. R. Ebling
W. F. Jerome
S. M. Martin
M. Nidd
M. R. Niemi
S. P. Wright

Caring for patients with chronic illnesses is costly—nearly \$1.27 trillion today and predicted to grow much larger. To address this trend, we have designed and built a platform, called Personal Care Connect (PCC), to facilitate the remote monitoring of patients. By providing caregivers with timely access to a patient's health status, they can provide patients with appropriate preventive interventions, helping to avoid hospitalization and to improve the patient's quality of care and quality of life. PCC may reduce health-care costs by focusing on preventive measures and monitoring instead of emergency care and hospital admissions. Although PCC may have features in common with other remote monitoring systems, it differs from them in that it is a standards-based, open platform designed to integrate with devices from device vendors and applications from independent software vendors. One of the motivations for PCC is to create and propagate a working environment of medical devices and applications that results in innovative solutions. In this paper, we describe the PCC remote monitoring system, including our pilot tests of the system.

INTRODUCTION

In 2003, the United States alone spent \$1.7 trillion on health care,¹ with more than 75 percent of these costs directed toward the treatment of patients with chronic diseases.^{2,3} Chronic diseases—which include diabetes, cardiovascular disease, chronic obstructive pulmonary disease, and cancer—are persistent or recurring conditions that require care for more than a year and that limit the patient's activities.³ Although there is no cure for a chronic disease, it can be managed to minimize its effects on the patient.

Today, patients with chronic illnesses typically monitor their condition at home through the use of simple medical devices such as a glucometer or

blood pressure cuff. Diabetic patients routinely test their blood sugar levels and often record them in a patient diary in order to monitor the effect of their food and medications. The patient diary is intended to help clinicians monitor the patient, but frequently it is too incomplete or illegible for effective use.

Because the risk of chronic disease increases with increasing age and is especially pronounced in those

©Copyright 2007 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of the paper must be obtained from the Editor. 0018-8670/07/\$5.00 © 2007 IBM

60 and over, these problems will be exacerbated in the coming years as the so-called baby-boom generation rapidly approaches this milestone.⁴ To

■ Today, patients with chronic illnesses typically monitor their condition at home through the use of simple medical devices such as a glucometer or blood pressure cuff ■

address these challenges, we must consider new mechanisms for managing chronic conditions. One such mechanism is remote patient monitoring.

Remote patient monitoring collects disease-specific metrics from biomedical devices used by patients in their homes or other settings outside of a clinical facility. Remote monitoring systems typically collect patient readings and then transmit them to a remote server for storage and later examination by health-care professionals. Once available on the server, the readings can be used in numerous ways by home health agencies, by clinicians, by physicians, and by informal care providers.

Consider the following futuristic scenario:

Tamsin Watson is an 86-year-old woman who lives alone. She was diagnosed with congestive heart failure and high blood pressure a number of years ago, and also suffers from seasonal allergy-induced asthma. When she was first diagnosed with congestive heart failure, her doctor prescribed a home monitoring system, called Personal Care Connect (PCC). The system came with a blood pressure cuff, a digital weight scale, and a data collection hub. With the help of PCC, every time she takes her blood pressure or steps on the scale, the readings are automatically transferred into her electronic medical record (EMR), where they can be analyzed by the system and reviewed by her physicians.

Once, when her weight was significantly above normal, the nurse called to find out how she was doing. In the course of the conversation, the nurse determined that she had gone to a celebration with her family and returned with a bunch of leftover

cold cuts, which she had been eating in sandwiches over the past two days. After discussing the situation with the doctor, the nurse recommended that she stop eating the sandwiches. She explained that cold cuts have a high sodium content, which was likely causing her to retain more fluids. The nurse called back the next two days until her weight had returned to normal.

After she experienced an acute instance of allergy-induced asthma, her pulmonologist added a spirometer (which measures air entering and leaving the lungs) to her home setup. When the system detects a decrease in her spirometer readings, it alerts her to alter her asthma medication in a manner directed by her physician. The latest system is said to incorporate allergen predictions to help patients decide when to begin asthma medications that take a few days lead time to become effective.

After she moved to an independent living facility, her PCC system started collecting sensor readings from her home environment as well. A few weeks ago, the system detected that she had had a restless night and that she was still in bed at almost noon. The sensor readings showed she had tossed and turned in bed and exited and reentered her bedroom a number of times. The system also showed that she took twice the normal dose of blood pressure medication in the morning. When she walked to the bathroom, the analysis of the data generated by the sensors in her slippers indicated a major change in her walking gait. Consequently, the system generated an alert condition for Mrs. Watson. Her doctor and the monitoring medical personnel were notified, as was her neighbor. The neighbor, a longtime friend, was registered as a local contact to check on Mrs. Watson's condition in cases like this.

The friend found Mrs. Watson to be very disoriented; she solicited emergency medical care and called Mrs. Watson's doctor and family. The doctor reviewed Mrs. Watson's recent remotely gathered medical data by means of a secure access to the central server data and ordered medical tests to be given to Mrs. Watson upon her arrival at the hospital.

This abbreviated scenario illustrates some of the applications and solutions that are possible by remote health-monitoring systems. In this paper, we present PCC and our approach to remote patient

monitoring. Several pilot trials have been conducted to assess the comfort level of patients, the ability of the system to collect the data, and the value of the data as input to potential applications, and we discuss our experiences with these trials. PCC is a middleware infrastructure. To enable all the elements of the scenario above, additional data sources will need to be integrated and new solutions implemented.

PCC consists of a data collection component that captures biomedical sensor data in the vicinity of where the data is generated and forwards that data to a server. The server normalizes the data, stores it in a database, and presents the data to applications through an application programming interface (API). The PCC system represents a new source of data for clinical and medical applications. Because the data is automatically recorded directly from the patient's biomedical sensor devices, physicians see an accurate representation of the patient's day-to-day condition.

Remote monitoring systems are being developed by companies, and research on remote monitoring technologies is being performed by researchers. Some of these are highlighted in the section "Related works." PCC is a standards-based, open platform for remote monitoring solutions. The goal is to use standards and open APIs in all key interfaces in the system. In the absence of standards, we use software and systems architecture to create plug-in interfaces. Medical sensor device vendors can add their devices to the platform, and the system software can be augmented to integrate the devices. Application vendors can implement their applications so that they are independent of the device technologies. The result is a working environment of device vendors and independent software vendors that leads to innovative solutions. This leads to less development time, less development expense, and faster time to market.

The remainder of this paper is structured as follows. The next section, "Design and implementation," reviews the design decisions that were made and why we made them and describes the initial implementation of the system design. The section "Solutions and deployment" provides details of pilot solutions that have been built or designed on top of PCC. The section "Summary" explains the contributions of the paper. The section "Related work"

reviews and contrasts PCC and other systems. The section "Future plans" outlines the next steps we are taking to advance the design of PCC.

DESIGN AND IMPLEMENTATION

PCC is envisioned as an open-system platform that contains core functions which will be needed by any solution that collects biomedical sensor data and stores the data on a server for long-term persistence and analysis. This paper describes the initial version of the design and implementation. Future versions of PCC may incorporate additional core functions, standards, and open APIs.

A key goal for the system is to use standards and open system APIs (based on collaboration with technology partners and standards groups) at as many points as possible. This is the key point that differentiates PCC from other remote monitoring systems. The success of PCC is dependent on the innovation and contributions of a broad set of technology partners. Biomedical device vendors and health-care application vendors should be able to integrate their technology innovation in the PCC system platform.

Data collection design issues

We begin by discussing the design requirements of the data collection component of remote health-care monitoring. Any remote monitoring system must start with the biomedical devices that generate the person's biomedical data. Remote health-care monitoring systems present a number of challenging requirements in the area of local data collection:

- *Patient usability*—The system should be easy for patients to use. It should require minimal training and minimal maintenance. It should minimize power consumption to avoid the inconvenience of recharging. It should be portable so that patients can take the system anywhere they go.
- *Scalability*—The system should scale to support large numbers of patients and their associated care providers. With 90 million Americans suffering from chronic conditions, a successful remote monitoring system could face significant numbers of patients very quickly.
- *Reliability*—The system should collect and store patient data, even in the face of network failures. Patients may travel to places where network connectivity is poor. The system should cache

data in these situations for later transmission to the back-end system.

- *Extensibility*—The system should extend to a broad range of home monitoring devices by using a wide variety of data exchange protocols. Standards, though important, take time to develop.
- *Financial resources*—The cost per patient required to deploy and maintain such a system should be minimal.

The financial requirement leads directly to a three-tiered architecture in which all of a patient's

■ A key goal for the system is to use standards and open system APIs at as many points as possible ■

biomedical devices communicate their readings to an intermediate component, which in turn communicates readings to a back-end system, consisting of the server and related applications. We call the intermediate component the *data collection hub*, or *hub*. The devices can use low-cost communication techniques to communicate with the hub because it is in the local vicinity of the devices. The hub has the computational resources to access a wide area network. In this way, only one wide area connection is needed per patient in the three-tier architecture.

The hub also offers a natural place to support reliability. If the hub loses network connectivity to the server back end, it can cache incoming data from the biomedical devices for later transmission. By placing reliability support on the hub, we minimize the functionality required of the biomedical devices and consequently reduce their cost. Any failure of a biomedical device to transmit readings to the hub can be treated as a device failure. Such events can be reported either to the patient or to the back-end system, depending upon the usability model.

The hub must operate with multiple biomedical devices and must extend to new biomedical devices. The hub is the natural place to deal with the variety of data exchange protocols and data formats used by the various biomedical devices. Because there are no wired standards for connecting biomedical

devices, a hub would have to have multiple wired adapters to support the variety of biomedical devices needed by patients. In addition, because each patient will likely use more than one device, the usability of the solution would suffer from the use of wired connections to the devices. These observations lead to the design question of whether to use a single wired connection that supports multiple protocols or a personal area wireless technology for connectivity between the medical devices and the hub. In the absence of a wired standard that supports this requirement, we made the decision to use wireless connectivity.

From a usability perspective, it is important to minimize the power consumption of the hub and the biomedical devices as the devices are likely to be battery powered and, in some scenarios, the hub will be battery powered. The most efficient mechanism for the device is to turn off the radio except during data transmissions. This means the device must initiate the data transfer and the hub must listen, in low-power mode, for incoming transfers. This shifts the power use from the device to the hub. The hub designs that have so far been envisioned contain either a long-range transceiver, which would necessitate regular power recharges, or are connected to a fixed network and have access to a wired power supply. In either case, shifting the power consumption for connectivity from the biomedical device to the hub maximizes battery life on the device.

The hub can also be used to increase the scalability of the end-to-end system. If the server is overloaded and cannot serve connection requests, the caching function on the hub effectively smoothes out peak loads. In addition, the hub can transform received data to a common format to simplify data integration on the back end and minimize the work performed by the server.

The hub function must run on multiple platforms. Mobile phones are an ideal platform for active patients, whereas a personal computer with broadband connectivity might be a better platform for homebound patients. Our solution is to implement the hub function using a software standard that is easily ported to a variety of platforms so that hubs can be developed in response to the varying requirements of patients.

In summary, the hub is the central element of the data collection part of PCC, interfacing with the devices and also with the server. Personal-area wireless technology provides the connection between devices and a hub that is portable to a number of platforms. The implementation of the hub must allow new devices to be added to the PCC system easily.

Data collection implementation

This section discusses the initial implementation of the data collection component of PCC, with particular reference to the design issues addressed in the previous section.

Personal-area wireless connectivity

We chose Bluetooth** as the wireless technology for communication between the biomedical devices and the hub. Bluetooth is a widely supported standard for short-range wireless communications. The standard was designed to enable low-power, low-cost, small-form-factor Bluetooth chipsets. The most commonly used class-2 Bluetooth radios have a range up to 30 feet.

Bluetooth is widely used in consumer electronics and is a common feature in notebook computers, personal digital assistants (PDAs), and some cell phones. At the end of 2005, there was an installed base of over 500 million Bluetooth devices,⁵ with 9.5 million units being shipped each week.⁶

Bluetooth provides several power management options. It supports a master-slave star topology, but also allows role switching. Consequently, the hub can both initiate and accept connections.

Over 30 Bluetooth connection profiles have been defined, including the popular Serial Port Profile. It is quite likely that any wired protocol being used by a device vendor will either fit directly with a Bluetooth profile or can be modified easily to use another existing profile.

Hub software

The software running on the hub is organized as shown in *Figure 1*. In implementing the hub software, we chose to use Java** 2, Micro Edition (J2ME), Connected Limited Device Configuration (CLDC),⁷ and Mobile Information Device Profile (MIDP).⁸ A broad range of devices, from cell phones to notebook computers, support this Java profile.

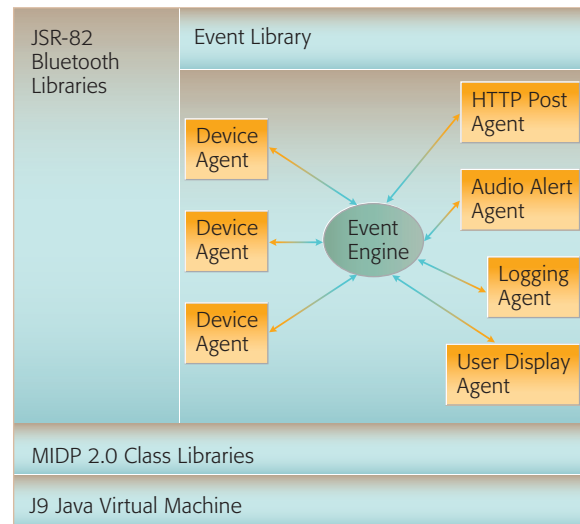


Figure 1
Hub software architecture

MIDP is the smallest standard Java execution environment. Both Sun Microsystems, Inc. and IBM offer compatible Java Virtual Machines and MIDP libraries. We chose Java as the implementation language because of its portability. The basic interface from Java to the Bluetooth hardware is provided by the JSR-82 libraries, the official standard Java API for Bluetooth.⁹ As the use of JSR-82 replaces the various custom Java interfaces to Bluetooth, applications will be able to run in many environments comprising Java and Bluetooth.

The core of the hub is an event engine that presents a blackboard communication model to application components, called *agents*, that allows agents to both subscribe to events and publish events. (In a blackboard communication model, an entity places a message in a commonly accessible area where other entities may access and read it if they so choose.) Device agents are essentially “device drivers” for sensor devices. They are specific to each type of medical device and capture incoming data by interacting with the device through the protocol used by the device. The device agent then uses the event library to convert the data into a generic event definition. A device agent must be able to connect to and exchange data with the device, identify correct and spurious data exchanges, and generate acknowledgments to the device.

A single event-description format is provided by the event libraries. This library comprises a hierarchical

framework of event definitions, serialization, and user extensions. Explicit serialization is required because MIDP does not support Java serialization.

■ To simplify event data access for application programmers, the PCC data access API uses a set of wrapper classes around the CEI programming interface ■

Some examples of current event definitions are: blood pressure, weight, pulse, oxygen saturation, and time stamp.

The Hypertext Transfer Protocol (HTTP) post agent is responsible for the delivery of events to the server. The agent sends the data to the Internet Protocol (IP) address and port contained in the configuration data associated with the MIDP platform. The HTTP post agent uses stable storage to cache events that cannot be sent immediately and forwards them when the next connection to the server is established.

A small number of other agents have been implemented. The audio alert agent makes preconfigured audio alerts in response to certain events, such as the reception of data from a device or successful transmission of an event to the server. The user display agent enables the display of status information and provides input capability for the user. The logging agent is a configurable logger that receives and sends message and trace-log events to the server for debugging purposes.

Hub device

The data-collection hub device requires both short-range wireless technology and Internet-based connectivity. One mobile device that meets those criteria is a Bluetooth-enabled cell phone with a wireless data service. The hub software was first implemented on a Sony Ericsson P900/P910 cell phone, but has since been ported to other cell phone platforms.

Biomedical devices

The initial version of the hub included device agents for an A&D Medical blood pressure cuff (Model UA-767 Plus), an A&D weight scale (Model UC-321BT),

and a Nonin Medical pulse oximeter (Bluetooth-enabled Avant** 4100 wrist-worn module).

Standards

We advocate the use of standards for sensor-to-hub protocols, medical sensor data formats, and representation of medical sensor data objects. However, until such standards are developed and adopted, we support vendors wanting to integrate their devices with PCC. A vendor with a Bluetooth-enabled biomedical sensor device must simply implement a device agent for the device. To facilitate this, IBM shares the event engine API, requisite hub libraries, and documentation with the vendor. If there is no existing event in the event library into which the data can be mapped, the event library may be extended. This design offers device vendors a very simple path to integrate their devices into PCC solutions until such time that standards are developed which make custom device agents unnecessary.

Server design issues

We now turn our attention to the server side of the remote monitoring system. Remote health-care monitoring systems present a number of challenging requirements for data management, user and device management, and deployment management. The requirements for the server side of the remote monitoring system include the following:

- *Extensibility*—The server architecture should be flexible so that the server can accommodate any hub device. The data model used by the server should support the rich variety of data that can be gathered from biomedical devices today, but should also extend to devices that are built in the future. Likewise, the API should offer flexibility to developers as they build applications to meet the evolving needs of clinicians and other care providers. Where appropriate, the interfaces and data model should apply existing standards or adopt those that will be developed.
- *Scalability*—The server architecture should scale to large quantities of data. It should offer efficient storage and retrieval of the wide variety of event types.
- *Interoperability*—The PCC server should interoperate with a variety of health-care organizations to allow care providers from disparate organizations to coordinate patient care.
- *Manageability*—The server should be easy to install, deploy, and maintain. The system should

provide tools to assist solution developers in deploying and managing data collection software on devices and the hub.

- *Security and privacy*—The server should protect the sensitive data that it maintains and offer access only to individuals authorized by patients or their representatives.

Of these requirements, extensibility led directly to many of our design choices. Extensibility requires that the system be adaptable to multiple data sources, including those not yet invented. It requires that the server be capable of interacting with any data collection hub, whether on a mobile phone, a PC, or an Internet appliance. It requires that the server cope with data sent in any data format using any communication protocol.

To achieve these goals, our server uses a multiple-adaptor design in which each adapter is specific to a particular hub family. The hub-specific adapter interacts with a hub to receive biomedical sensor data, uses knowledge of the data format presented by that hub to extract data item elements, and prepares the data for storage in the server.

Once it has received data from the hub, the server then stores that data in a data repository. The data in the repository must have an extensible data format because multiple types of biomedical data exist today and because the medical field continues to evolve. Although some biomedical data can be represented as a single data value, other biomedical data may require an array of data values, and still other biomedical data may consist of a continuous stream of data. The data model used by the server should ultimately be based upon a widely accepted standard.

Once the data has been stored in the repository, the server must then provide access to that data. The APIs should support today's applications, but should also extend to future, unanticipated application requirements because this field is in its infancy. Like the data model, the API should also be built upon widely accepted standards so that any application can run on any remote monitoring server.

The entire server architecture must be scalable. A large percentage of patients with chronic conditions could need such a system. The underlying repository implementation must support contents that vary from small pilots of short duration serving a

small number of patients to major deployments lasting many years and serving the needs of many patients.

Another aspect of scaling is interoperability. Once remote monitoring is in widespread use, the system must allow interoperability among care providers and care organizations. The focus of the system should be on providing the best care possible for the patient. This implies that the various care providers and care organizations should be able to access the same information.

Like all systems, remote health-care monitoring requires security, privacy, and manageability. Security and privacy are perhaps more pronounced in the health-care industry due to the Health Insurance Portability and Accountability Act (HIPAA)¹⁰ of 1996 in the United States and similar regulations elsewhere, but the fundamental requirements are not unique. We discuss security and privacy from an end-to-end perspective later in the paper.

Server implementation

This server design deals with the functions considered to be fundamental for a remote monitoring server. More extensive functional enhancements and new functions can be added as the server evolves. We group our discussion into the two main components of the server: the data management component and the user-device and deployment management component. The system diagram of the PCC server that we implemented is shown in *Figure 2*.

Data management

The data management portion of the PCC server consists of four major components: a set of hub-specific adapters, a data normalizer, a data repository, and a data access API. The hub-specific adapter receives and processes biomedical sensor data forwarded by the hubs. The data normalizer then extracts data items from the events and stores the data in the internal representation used by the data repository. The data access API element provides access to the data repository and presents the data to application programs as Java objects.

Hub-specific adapters

The PCC server design supports multiple adapters. The multiple adapter design paradigm is critical to creating an open-PCC server platform. Each adapter is specific to a hub family. A hub family is

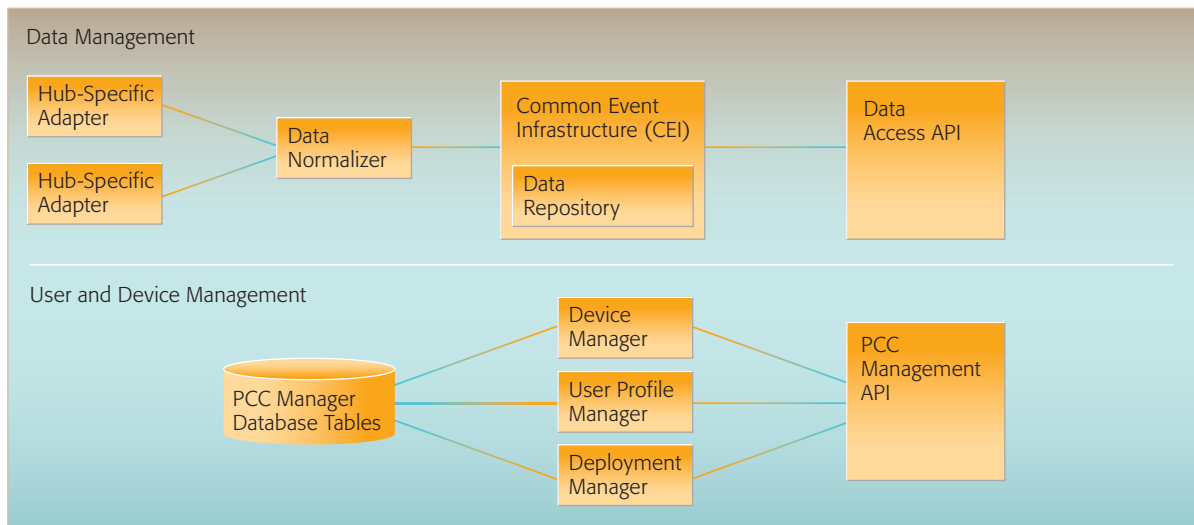


Figure 2
PCC server-side components

characterized by the following attributes: device-to-hub communication, event data representation, event serialization, and hub-to-server interaction. Different hub families can use different event classes, different communication protocols, and different serialization schemes. While the hub described in the earlier subsection “Hub device” is open, a vendor may want the freedom to implement things differently. A hub family may be optimized to handle a specific set of biomedical sensors. Each adapter is responsible for communicating with a family of hubs by using some protocol, deserializing the data forwarded by the hub, and extracting event data from the event after instantiating the events.

The hub family implemented as described earlier in the section “Data collection implementation” is called *hub family A*, and the adapter implemented in the initial version of PCC is *adapter A*. Hub A can send multiple events in a unit called an *envelope*. Adapter A deserializes the envelope, instantiates the events, parses the events for event data items, and passes the data items, per event, to the data normalizer component. The events can be biomedical sensor events or informational events, such as hub log records.

Data normalizer

The data normalizer receives events from the hub-specific adapters, normalizes that data for storage in the data repository, and stores the data in that

repository. The Common Event Infrastructure (CEI),¹¹ with its extensible and flexible Common Base Event (CBE),¹² is a key building block of the data management component of the PCC server. CEI is a middleware component for managing the life cycle of CBEs. The CBE is an extensible structure for event data that supports logging, system management, problem determination, autonomic computing, and e-business functions in an enterprise. Cisco Systems, Inc. and IBM collaborated on the specification that has been submitted to OASIS (Organization for the Advancement of Structured Information Standards). The CBE specification allows the creation of new fields with the field name and data type selected by the user; this is the main extensibility feature of a CBE. CEI has functions for the creation, deletion, storage, retrieval, and modification of CBEs. CBE is the common format for event data in the PCC server.

Once the data normalizer receives the event data from an adapter, it creates a CBE representation of the event and stores the CBE event in the CEI data repository. The extensibility features of CBE ensure the ability to map new biomedical events into the PCC server data format.

CEI-based data repository

CEI has both synchronous and asynchronous APIs to access the CEI repository and can store and retrieve CBEs in the form of Extensible Markup Language (XML) documents or Java objects. The

CEI API uses the concepts of Java Messaging Service (JMS),¹³ Enterprise JavaBeans** (EJBs),¹⁴ and XPath.¹⁵ To simplify event-data access for application programmers, the PCC data access API uses a set of wrapper classes around the CEI programming interface. The Java objects returned by the PCC data access API have a set of common attributes: a patient identifier, the time stamp of the event, and the name of the event. The PCC data access API also supports both synchronous and asynchronous methods to retrieve event objects and a method to delete event objects.

In the long term, the PCC data access programming interface (methods and objects) should be based on a standard. The current implementation of the data access API should be considered a starting point to develop such a standard. We considered using the Clinical Document Architecture (CDA) of HL7**¹⁶ but decided against doing so due to the storage and computation overhead of manipulating CDA/HL7 documents. Most clinical information systems use an efficient representation of the data for local applications and use CDA/HL7 representation of the data for interoperability.

User, device, and deployment management

To support system manageability, key server-side systems management functions were implemented. These include a user profile manager, a device manager, and a deployment manager. These elements are ideal for small- to medium-scale engagements. For larger engagements, more sophisticated function and scalability will be needed. Enterprise software companies have product offerings in these areas. At the end of this section, we discuss the issues of security and privacy.

User profile manager

The PCC user profile manager maintains a unique PCC identifier and basic contact information for each patient and provider. We recognized that, upon adoption of this solution, a hospital or health-care provider would already have a comprehensive patient and provider database. Once adopted, the solution would need to get its information from this database. Consequently, we capture only the very basic patient enrollment information.

Device manager

The PCC device manager maintains information on the hubs and devices and their relationship to the enrolled patients. This is accomplished through a

notion of *kits*, where a kit is defined as the collection of one patient, one hub, and any number of necessary biomedical devices. The device manager allows the creation of kits and the management of attributes (metadata) of the hub and devices within that kit. This information is thus created and maintained in the device manager database.

Deployment manager

The PCC deployment manager provides some functions to assist in the automatic deployment of the MIDP application and its JAD (Java application development) file. The JAD file contains the configuration data needed by the application.

The deployment unit is downloaded to the patient's hub by means of a user interface (Web page) on the hub. On the Web page, the user provides a unique identifier and password and forwards the request to the deployment manager by using a specific URL. The deployment manager interacts with the PCC user profile manager and the PCC device manager to produce a JAD file, which is sent as the response to the request. Based on the MIME (multipurpose Internet mail extensions) type of the downloaded JAD file and the contents of the JAD file, the phone downloads the J2ME MIDlet (as a JAR file) from a location on the server.

Security and privacy

Maintaining the security and privacy of a user's biomedical data is necessary to meet security requirements, user expectations, and regulatory compliance. A set of coordinated approaches is needed to secure the data as it moves from the device to the data collection hub to the server to the application. Because PCC is an element of an overall solution, some of the security mechanisms are implemented within PCC, and others are a part of other solution components.

Few personal biomedical devices have features to secure the data. Because the devices are normally in the physical possession of the user, unencrypted data on the device is not a major security threat. As the devices become more sophisticated and hold biomedical data that is more sensitive, device vendors may move to on-device encryption to protect the data. The first major point of insecurity in today's systems is the transmission of data over the wireless link to the data collection hub. This threat is of a type common to Bluetooth applications

and has been addressed directly by the security provisions of the Bluetooth protocol definition.¹⁷ The pairing, authentication, and link encryption that are integral to Bluetooth, when used as recommended, provide adequate security to this connection, which is generally located within a person's home.

Data may reside on the hub for some time before it is sent to the server. A security threat is created if the data is stored in the clear on the device. This security problem is resolved if the hub supports encryption of data cached on the hub. The current version of PCC does not encrypt data on the hub, but adding support for such encryption would be straightforward. If strong security on the hub is required, it may be necessary to use a customized hub platform. The use of a customized hub platform would not violate the open platform goal of PCC as long as the hub has open APIs to integrate biomedical sensor devices.

A number of techniques can be used to secure communications between the hub and the server, including Secure Socket Layer (SSL), Secure HTTP (HTTPS), and virtual private network (VPN). PCC has been tested with a VPN based on the IBM WebSphere* Everyplace* Connection Manager (WECM) using HTTPS secure posts. We have not tested PCC with an SSL-based solution.

In addition to ensuring secure hub-to-server communication, it is also necessary to ensure that the communication is reliable. Although several reliable enterprise messaging solutions exist, they are based upon PCs, rather than less sophisticated, more portable devices. The PCC hub sends the events to the server in the same time order in which they were received from the devices. The events are sent in an HTTP post. The server does not generate an HTTP success response until the event is successfully written to the event repository. The hub does not delete an event from nonvolatile storage until it has received the HTTP success response. If the server does not respond or if the server sends a nonsuccess HTTP response, the hub retries sending the event.

In many of the solution scenarios, the applications that query the PCC event repository are Web-accessible. In these scenarios, PCC is a component of a networked service or solution. The PCC-based network service is thus vulnerable to a set of

security threats common to networked services, such as denial of service attacks, session hijacking, malware (hardware, software, or firmware that is intentionally included or inserted in a system for a harmful purpose), authentication hacking, and network intrusion. Security solutions exist to address those threats, but further discussion is beyond the scope of this paper.

If the security of a PCC-based network is breached, the patient's biomedical data is vulnerable unless security precautions are taken. The event data consists of event attributes and data values. The event attributes are the patient identifier, event time stamp, and event name. Unless one has the mapping of the patient identifier to the patient's name, the patient cannot be identified. To further protect the event data, we have designed and plan to implement encryption and decryption of the data values of the event. These values are not pertinent to the PCC query mechanism, as only the patient identifier, the time stamp, and the event name are used in searching the PCC repository for relevant data. Once this additional security is implemented, someone who gains access to the service data would only be able to learn information like the fact that a blood pressure event was recorded at 14:25 for patient with identifier 1234567; they could not learn the actual data values for the blood pressure reading.

In the initial version of PCC, privacy management is handled by the solution; there is no privacy control support in PCC. Future versions of PCC can incorporate the identity of the requester in each data access request. An integrated or external-to-PCC privacy management system can then resolve the question: "Is requester X allowed to gain the requested access to data of patient Y?"

SOLUTIONS AND DEPLOYMENT

To better understand the requirements of pervasive health care and to test the PCC infrastructure, we designed and deployed some solutions. The solution at the University of Heidelberg is described in another paper published in this special issue.¹⁹ Another solution at an early stage of definition is the Healthcare@Home system. Healthcare@Home is a two-year project funded by the United Kingdom Department of Trade and Industry. The project aims to provide extensive remote monitoring of patients using medical sensors, integrate the data from other health-care systems, and analyze the data using grid

computing. The focus disease will be diabetes. There are plans to develop a demonstration prototype. Two additional solution pilots are described below.

Arhus Eldercare Project

IBM, the municipality of Arhus, Denmark, and the University of Arhus collaborated on a project whose goal was to assess how the municipality could use remote monitoring technology to empower and assist the self-care of elderly citizens who lived in their homes or in assisted-living facilities. The municipality and national government are responsible for the health-care services for all citizens. Currently the municipality uses visitation to monitor the elderly who are at risk in their homes.

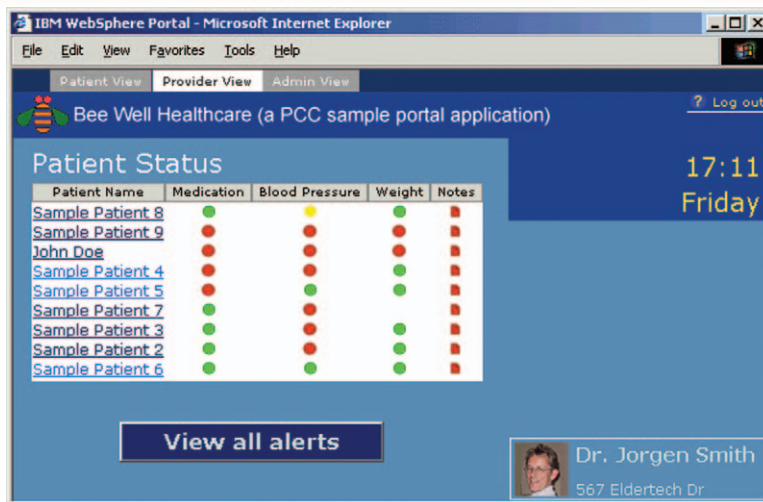
The initial pilot operation began with six elderly Arhus citizens, ages 75–90, for a period of three months. Each citizen was given an A&D blood pressure cuff, an A&D scale, a data collection hub (cell phone), and a tablet PC. The citizens were asked to use these devices to measure their blood pressure and their weight on a daily basis. In addition, they used the tablet PC and a portal application to record their medications. The PCC system made the collected data available to care providers. The key requirements that were presented by this pilot and had to be satisfied included the following:

- *Data analysis*—At the request of the municipality, the Arhus pilot added an analysis component as a PCC application. Specifically, we implemented an analysis for each of the event streams (blood pressure, weight, and medication) to determine medically significant events. The analysis routines generated citizen status updates and notification messages, which were viewed by the municipality medical professionals by using the portal component of the solution. The citizen could also view this information.
- *Integration with the existing environment*—Another requirement was to integrate, in selected areas, with the provider's preexisting environment. This environment included a legacy health-care application that manages the citizens' prescriptions and the replication of a paper logbook normally kept in the citizen's home. The municipality requested that we detect changes in medication prescriptions and integrate the changes with the medication analysis. A link was

established with a legacy management system, and the prescription records of the pilot participants were updated in a separate prescription database that was established for the pilot. The medication analysis used the prescription database and performed checks for incorrect medication and missing medication for the citizens in the pilot. Before this solution, citizens maintained a paper logbook that recorded, among other things, their medications. We created an electronic, Web-accessible logbook that made it easier to administer the care of the citizens.

- *Rich applications*—Another goal of the pilot was a set of functionally rich, Web-based applications for citizens, approved friends, family members, and providers. In the first phase of the pilot, we explored functions for citizens and providers. (The views illustrated in **Figures 3A** and **3B** are very similar to the application views used in the pilot.) One of the provider views is a dashboard that has a summary view of each citizen (Figure 3A); red, yellow, and green indicators are used for medication, blood pressure, and weight. The column labeled "Notes" gives a visual indicator for the existence of and type of unacknowledged notifications. Each citizen's name is a link that takes the provider to the patient view (Figure 3B), which shows details for that citizen. Another link on the citizen dashboard view takes the provider to a view of all the unacknowledged notifications. Acknowledgments can be performed through the portal application. The patient view is used by the citizen and providers to show information about a citizen. A citizen can access information about himself or herself. The information includes a list of medications and reminders to take the medications, and it provides a means for the citizen to acknowledge taking the medication. Citizens can also view their medical event history. In addition, the patient view provides access to interdisciplinary notes and notifications and a visitors logbook.
- *Status and lessons learned*—The first lesson learned was that the analysis results must be succinctly presented to care providers, and they must be summarized into a small number of notifications per citizen per day. We experienced an issue with notifications for missing medications—medications that were not taken on time according to a set schedule. Some citizens have up to six medications, and some medications have to be taken six times a day. The medication analysis

A



B

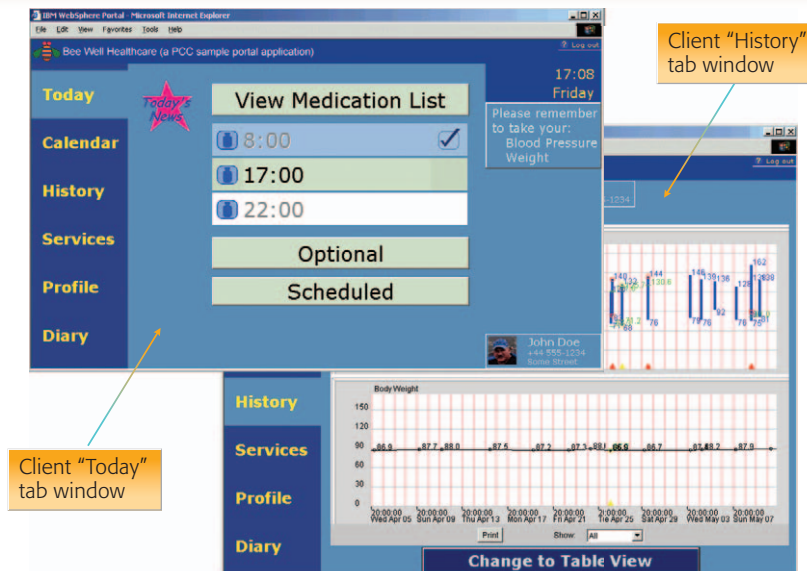


Figure 3

Functional views: (A) citizen dashboard view, (B) patient view

routine generates a “caution” notification if the citizen is over one hour late taking a medication and an “alert” notification if the citizen is over four hours late taking a medication. These medication rules were designed to provide sufficient information to make timely decisions; however, the side effect was a large number of notifications. Because medication events are registered using a portal application, missing medication notifications accumulate if the citizen forgets or is not home. This problem has been

partly addressed through improved functions for management of notifications.

The portal application that was used to register medication events showed the importance of extensibility in the PCC architecture. In particular, this pilot showed that, in addition to medical devices, applications could be viewed as possible data sources to PCC.

Another challenge that we encountered was citizen authentication to the portal server. A number of

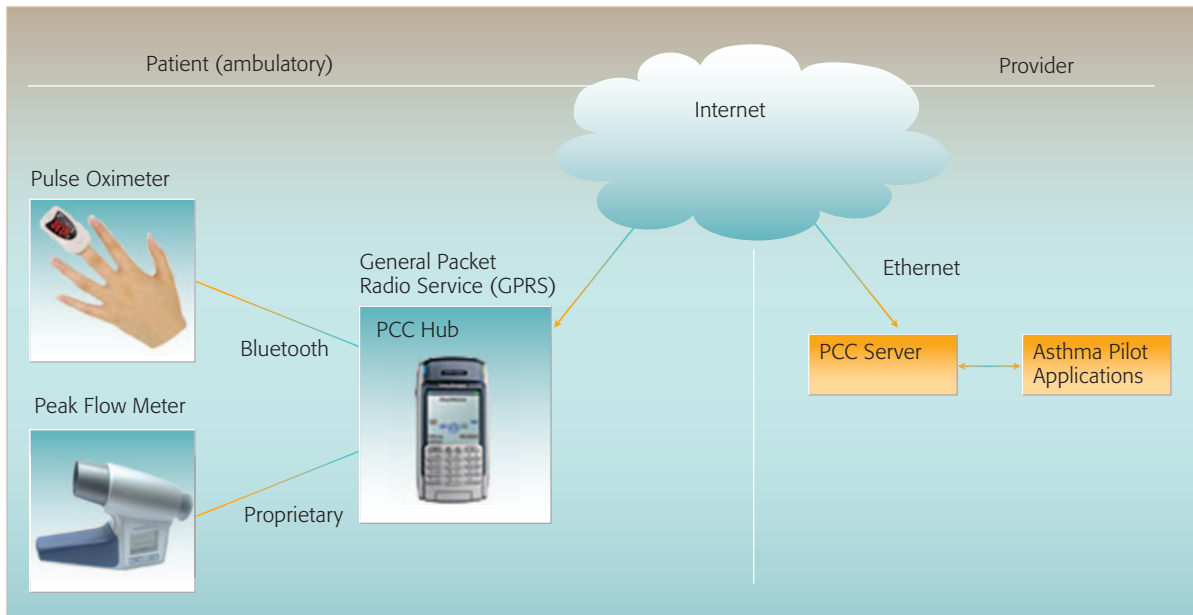


Figure 4
Asthma pilot

applications run on the tablet PC, including registering medication events. Citizens must log in to the portal server several times a day. The main user input device for the citizen on the tablet PC is the tablet pen because none of the citizens in the pilot was familiar with a keyboard. Entering the password presented a problem to some users. This problem can be eliminated in the future with biometric authentication or a Bluetooth-enabled pill dispenser that automatically forwards pill-removal events.

PCC asthma study

The PCC asthma study is a proposed collaboration between a university medical research institution and the IBM Healthcare and Life Sciences (HCLS) solution group. The plan is for IBM to build the solution and the university researchers to do the clinical study. Although all details of the collaboration have not been finalized, some aspects of the plan are presented below.

Asthma is one of the chronic diseases that affects all segments of the population. The initial patient group will use devices in the home environment. The patient groups will be expanded later to include mobile workers, college students, and even young children. The first phase of the project will focus on the use of technology to gather data from a small

number of patients. The second phase will include a larger number of patients and focus on clinical research outcomes. Currently, patients treated by the research group use a peak flow meter to measure lung function and a paper-based diary to record how they feel at the time. The peak flow meter reading is recorded in the diary entry. The patients bring the diary to periodic office visits.

In the proposed study, the scope of the collected patient data will be expanded, and the timeliness of the data will be improved. The solution components (*Figure 4*) are listed below:

- Devices: Pulse oximeter and peak flow meter
- PCC hub: Mobile phone running both the hub software and the diary-capture application
- PCC server
- Asthma pilot analysis and user-interface application

The pulse oximeter, already incorporated into PCC, will be used to capture and send the patient's pulse and oxygen saturation level to the hub. Oxygen saturation level is an additional indicator of lung function. Patients will continue to use the peak flow meter that they are currently using. However, instead of recording these readings in a paper diary, patients will use an electronic patient diary (EPD)

application to record this information, together with information about how they currently feel. The EPD application will be supplied by a third-party vendor. Once a Bluetooth-enabled peak flow meter becomes available, it can be incorporated into the system, and patients will no longer be required to enter their peak flow readings manually.

The asthma pilot analysis and user interface application enables processing and display of the remotely captured patient data to the provider. The proposed interface includes a patient dashboard that

■ To support system manageability, key server-side systems management functions were implemented ■

the provider can use to monitor the data of the patient. The dashboard will display the data in a summary view and will designate compliance or noncompliance with the guidelines established for the patient. Should a patient's data fall outside the acceptable range, the dashboard will indicate this, and the provider will then be able to view detailed data related to the patient.

Medical researchers expect to see major improvement in the effectiveness of treatment as a result of moving from paper-based, retrospective diaries to electronic capture of diary data. Some researchers reported observing patients filling out their diaries in the waiting room. It is hoped that the ease of using the EPD application will lead to more accurate and timely diary information. Because the diary entries will be time-stamped, the researchers will be able to correlate the patient's diary input with data collected at about the same time. Because the diary information will be in electronic form, it can be analyzed for trends. Researchers will have timely access to the diary information instead of waiting for the patient's next office visit. By being provided with more timely, accurate, and frequent patient information, the clinicians can be more proactive in their treatment of the patient.

The flexibility of the PCC architecture and design is key to integrating the EPD application into this solution. Further, the EPD represents an additional application as a data source (rather than a mon-

itoring device), much like the medication compliance application in the Arhus solution. In the asthma study solution, the EPD is integrated as a data source on the hub rather than a data source directly to the PCC portal.

RELATED WORK

PCC is not the first system that targets the remote monitoring solution space. Examples of companies offering commercially available remote monitoring systems are Honeywell, Health Hero Network, American Telecare, and AMD Telemedicine. The systems offered by these companies are proprietary; their hubs work only with that company's sensors, their servers interact only with their hubs, and their applications and services have proprietary access to the sensor data. In contrast, PCC is an extensible hub-server platform for remote monitoring solutions.

Remote health-care monitoring has also been an area of exploration by numerous researchers. Yao et al.²⁰ have explored the use of standards in two key points in the data flow. They explore a plug-and-play system for home medical devices based on the IEEE 1073 Medical Information Bus standard. This standard was proposed for in-hospital medical devices and has not been widely accepted among vendors who market less sophisticated home medical devices. The lack of standards in home medical devices is the major reason PCC pursued its software approach. The emphasis on standards is common to both PCC and the work of Yao. In Lebak et al.,²¹ the same research team describes a monitoring system that collects biomedical data from patients in their homes and forwards the data in an HL7-compliant format to the patient's EMR, located at a remote site. The complexity of HL7 necessitates the use of a PC as a hub and a vendor application to perform HL7 processing. The work of both Yao and Lebak fails to address a number of the extensible remote monitoring platform issues introduced earlier in the section "Design and implementation."

Other researchers have focused on novel sensor systems for mobile users and have built an application platform to interface to the sensor system. The problem is that the platform is not extensible to other types of sensors. Chen et al.²² describe a system that supports the mobile patient by using wireless sensing devices and a mobile

cellular phone. They use specialized sensors to gather four sensor streams and then use an integrated processor to generate 10 other biometrical streams. Cárdenas et al.²³ describe a system that collects data from a VivoMetrics LifeShirt**²⁴ and presents it visually. Both the Chen system and the VivoMetrics system support continuous monitoring of ambulatory patients. These systems are complementary to PCC because they can be adapted easily to PCC either by using a device component in the hub or by extending the server to treat the sensor system as a distinct hub.

Cellar et al.²⁵ describe a remote monitoring solution, called Home Telecare Systems, that uses home medical equipment connected to a PC in combination with a wireless ambulatory device that supports a panic button and an accelerometer to detect stumbles and falls. The combination of medical sensors and activity sensors is a future direction for PCC, but PCC will incorporate activity sensors as extensions to its open platform and not as integrated elements of a proprietary system.

Other researchers have looked beyond such basic devices as cell phones and PCs to provide hub services. Ling et al.²⁶ describe an interesting approach to home monitoring that uses a robotic pet as the data communications hub. In addition to supporting remote health-care monitoring, the robotic pet can also act as entertainment and as a “companion” to the homebound patient. Paggetti and Tamburini describe the Telecare Platform,²⁷ which was deployed to 2,500 users in Tuscany. Like PCC, this system uses an architecture that incorporates a central communications hub. However, because the Telecare Platform hub is instantiated as a set-top box unit, users are limited to using the system in their homes and cannot travel as easily as they can with PCC. The goal for the Telecare Platform is wider than that of PCC in that it includes support for home automation solutions and can provide access to entertainment and social and educational services—all features not included in the PCC solution. No mention is made about whether the Telecare Platform uses an open architecture nor whether the platform uses wired or wireless remote biomedical monitoring devices.

Bar-Or et al.²⁸ describe a system called BioStream that offers real-time analysis of continuous biomedical sensor streams and provides persistent

storage of those data streams. BioStream is an integration of an analysis framework and a stream data collection facility. PCC can support stream

■ As the field evolves and key standards become better defined and adopted, our architecture can make use of these standards ■

processing of discrete events through its asynchronous data access methods. The BioStream paper focuses on analysis of continuous sensor streams and does not mention extensibility of its platform, while PCC emphasizes the extensibility and openness of the PCC platform. The analysis framework portion of BioStream is complementary to PCC, as PCC currently does not have an analysis infrastructure.

Lorincz et al.²⁹ present a sensor network for responding to emergency situations. Their work, called CodeBlue, focuses on building an information plane in which a diverse set of medical sensors, based upon mote technology,³⁰ can discover one another and communicate as necessary. Unlike CodeBlue, PCC focuses on collecting data from commercially available home-based monitoring devices and presenting that data appropriately to a variety of health-care providers. Like Lorincz, Amato et al.³¹ present an approach to remote health-care monitoring based upon wireless sensor networks using Crossbow Technology MICA2 motes. Although such a solution is technically possible, in our opinion it is less practical than alternatives such as PCC or the many home-based monitoring products. In both cases, the “last mile” involves a wireless set of sensors that monitor the vital signs of a patient. The difference between the two approaches is in how data is transmitted to the servers that ultimately store the data. With the Amato approach, a network of sensor nodes would forward the data along the path toward the server. With the PCC and similar approaches, a communications hub forwards the data to the server using wide-area connectivity (through either a wired or a cellular data network). For the foreseeable future, the assumption of wide-area connectivity is more practical than the assumption of a wide-area sensor network that provides sufficient connectivity.

SUMMARY

PCC is a standards-based open (client and server) platform for interfacing to a wide variety of biomedical devices and sensors, collecting data from the devices and sensors, storing the data in a server repository, and making the data available to applications through a documented API. We advocate the use of standards for all key interfaces and

■ We are exploring ways to extend PCC to serve as a node in an interoperable health-care network, such as NHIN ■

APIs. In the absence of key standards, we have used software architecture on the hub and an adapter-based architecture on the server for extensibility. As the field evolves and key standards become better defined and adopted, our architecture can make use of these standards.

The communications link between the devices and the hub is the standard short-range wireless Bluetooth. Any other nonproprietary wireless technology with similar characteristics can also be used. We have designed and implemented a hub family; the hub is implemented as an application using standards-based J2ME MIDP 2.0. The hub was specifically designed with a software architecture that enables device vendors to interface to the hub in a straightforward manner. Although the hub described in this paper uses a cell phone, it could be ported easily to any platform—such as a PDA, tablet PC, or regular PC—that has a J2ME MIDP environment.

The main features of the server are the following:

- *A set of adapters that interface to hub families*—The adapter should be viewed as a server plug-in that makes it possible for other hubs, even those from other vendors, to plug into and use the server infrastructure. Sensor vendors can integrate with the hub or with the server through an adapter.
- *An event data repository*—The repository data model is based on the CBE,¹² which has been proposed as a standard by a group that includes IBM and Cisco Systems.
- *A data access API for retrieving the event data*—The API has methods for synchronous and asynchronous access and uses Java objects to encapsulate the event data.

We have described the implementation of one pilot and the design of another pilot. In addition, a second completed pilot is described in detail in another paper in this issue.¹⁹

Our push for standards and open platforms is an attempt to spur a collaborative network of medical sensor vendors and application developers. Medical sensor vendors can use our platform to integrate their devices to gain seamless connection to an API available to numerous applications. Clinical-information system developers can have a uniform view of a wide variety of medical sensor data without the need to know how to connect to each new device that comes on the market. Both groups will be free to innovate in their areas, knowing that it contributes to the end-to-end solution.

FUTURE PLANS

The biomedical data collected by PCC is a source of data that has not been fully integrated into patient care. In the PCC pilots, Web portals are used to give patients and providers access to the data with a variety of analysis support. Efforts are underway to integrate PCC more closely into clinical systems and to make the data available through interoperable health-care networks. We are exploring ways to extend PCC to serve as a node in an interoperable health-care network, such as the National Health Information Network (NHIN).³² A regional health-care prototype based on the IHE (Integrating the Healthcare Enterprise) Cross-Enterprise Document Sharing (XDS) profile has been implemented; the goal is to integrate PCC as an operating node in that network.

Today, the open platform vision for PCC is achieved by our willingness to share internal APIs and by the use of technologies that serve as the basis for a standard. The necessary standards are not in place; consequently, some focus must be placed on enlisting technology partners and standards organizations to develop them. The first area to focus on is a standard way of interconnecting medical devices and data collection hubs. There has been some activity in the Bluetooth Special Interest Group (SIG) (the body responsible for defining the Bluetooth standard). In response to general market

demand, the Bluetooth SIG has formed a medical devices working group to define how medical devices should communicate over Bluetooth, with consideration given to the requirements of device manufacturers, end-to-end solution providers, health-care payers, and others. In addition, the European Standardization Committee has developed a standard for the representation of vital signs information.³³ Although this standard is intended to enable interoperability among different manufacturers' devices within a hospital setting, the data format may be amenable to use in remote monitoring environments, such as PCC.

On the server, data models and data access API standards will provide the uniformity sought by application developers. A standard data model would define how sensor data is represented in storage and in the message formats of the sensor data. The data access API standard would define the representation of data as viewed by applications and define the set of method calls to access the data. As medical sensor technology evolves, new forms of sensor data will be developed, and active standards groups will need to map the new sensor data into these standards. PCC currently uses the CBE,¹² which can serve as a starting point. Just as Structured Query Language (SQL) jump-started the widespread acceptance and use of relational databases, data-model and data-access standards will likely play a similar role in the acceptance and use of remote-monitoring applications and solutions.

An area of great interest is extending PCC to support monitoring the condition of elderly patients living alone, or even within an assisted-living environment. A number of researchers have been exploring how to monitor the activities of daily living.^{34,35} We believe the PCC solution can be extended to support environmental sensors and can integrate that data with the monitoring of vital signs. If remote monitoring technologies can help an elderly person live independently for a longer period of time, the elderly patient can maintain a higher quality of life and the community can save care dollars.

Finally, as remote monitoring becomes widespread, we anticipate that clinicians and other care providers will be overwhelmed with data. Thus, an important aspect of future work will be in exploring how to best represent this data to them. Another important challenge is to automatically identify important medical events that should be brought to

the physician's attention. These challenges represent active areas of our current research. As technology improves, it will focus on helping patients monitor their own conditions.³⁶

ACKNOWLEDGMENTS

We thank the many people who contributed to the design, implementation, and pilot studies associated with Personal Care Connect, including Joseph Jasinski, Jason Jho, Shyjee Mathai, Kathy Schweda, and Edie Stern. We also thank Michael Svinte, Kenneth King, and Edward Macko for their executive support.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of Bluetooth SIG, Sun Microsystems Inc., Health Level Seven, Inc., Nonin Medical Inc., or VivoMetrics, Inc. in the United States, other countries, or both.

CITED REFERENCES

1. Health Insurance Cost, National Coalition on Health Care, <http://www.nchc.org/facts/cost.shtml>.
2. Chronic Disease Overview, Centers for Disease Control and Prevention, <http://www.cdc.gov/nccdphp/overview.htm>.
3. *State Officials' Guide to Chronic Illness*, The Council of State Governments, <http://csg-web.csg.org/pubs/Documents/SOG03ChronicIllness.pdf>.
4. *Long-Range World Population Projections: Based on the 1998 Revision*, The United Nations Population Division, POP/DB/LRP/Rev.1998.
5. Bluetooth Technology Benefits, Bluetooth SIG, <http://bluetooth.com/Bluetooth/Learn/Benefits/>.
6. *Bluetooth Shipments Double in Four Months*, Bluetooth SIG, http://bluetooth.com/Bluetooth/Press/SIG/Bluetooth_Shipments_Double_in_Four_Months.htm.
7. Java ME: Connected Limited Device Configuration (CLDC); JSR 30, JSR 139, Sun Microsystems, <http://java.sun.com/products/cldc/>.
8. Java ME: Mobile Information Device Profile (MIDP); JSR 37, JSR 118, Sun Microsystems, <http://java.sun.com/products/midp/>.
9. JSR 82: Java APIs for Bluetooth, Sun Microsystems, <http://www.jcp.org/en/jsr/detail?id=82&showPrint>.
10. *Health Insurance Portability and Accountability Act (HIPAA) of 1996*, Public Law 104-191, U. S. Department of Health and Human Services, <http://www.hhs.gov/ocr/hipaa/>.
11. Common Event Infrastructure, IBM Corporation, <http://www-306.ibm.com/software/tivoli/features/cei/>.
12. Common Base Event, IBM Corporation, <http://www-128.ibm.com/developerworks/library/specification/ws-cbe/>.
13. J2EE: Java Message Service (JMS), Sun Microsystems, <http://java.sun.com/products/jms>.

14. Java EE: Enterprise JavaBeans Technology, Sun Microsystems, <http://java.sun.com/products/ejb/>.
15. *XML Path Language (XPath), Version 1.0*, J. Clark and S. DeRose, Editors, W3C Recommendation (November 16, 1999), <http://www.w3.org/TR/xpath>.
16. *HL7 Clinical Document Architecture, Release 2.0*, R. H. Dolin, L. Alschuler, S. Boyer, C. Beebe, F. M. Behlen, P. V. Biron, and A. Shabo (Shvo), Editors, Health Level Seven, Inc., <http://www.hl7.org/v3ballot/html/infrastructure/cda/cda.htm>.
17. Bluetooth Wireless Security, Bluetooth SIG, <http://www.bluetooth.com/Bluetooth/Learn/Security>.
18. S. Garfinkel, *Web Security, Privacy and Commerce*, O'Reilly Media, Sebastopol, CA (2002).
19. C. Kirsch, M. Mattingley-Scott, C. Muszynski, F. Schaefer, and C. Weiss, "Monitoring Chronically Ill Patients Using Mobile Technologies," *IBM Systems Journal* **46**, No. 1, 85–94 (2007, this issue).
20. J. Yao, R. Schmitz, and S. Warren, "A Wearable Point-of-Care System for Home Use That Incorporates Plug-and-Play and Wireless Standards," *IEEE Transactions on Information Technology in Biomedicine* **9**, No. 3, 363–371 (2005).
21. J. W. Lebak, J. Yao, and S. Warren, "HL7-Compliant Healthcare Information System for Home Monitoring," *Proceedings of the 26th Annual International Conference of IEEE Engineering in Medicine and Biology Society*, San Francisco, CA (2004), pp. 3338–3341.
22. W. Chen, D. Wei, M. Cohen, S. Ding, S. Tokinoya, and N. Takeda, "Development of a Scalable Healthcare Monitoring Platform," *Proceedings of the 4th International Conference on Computer and Information Technology*, Wuhan, China (2004), pp. 912–915.
23. A. F. Cárdenas, R. K. Pon, and R. B. Cameron, "Management of Streaming Body Sensor Data for Medical Information Systems," *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Science*, Las Vegas, Nevada (2003), pp. 168–191.
24. LifeShirt System, VivoMetrics, Inc., <http://www.vivometrics.com/site/system.html>.
25. B. G. Celler, N. H. Lovell, and J. Basilakis, "Using Information Technology to Improve the Management of Chronic Disease," *The Medical Journal of Australia* **179**, No. 5, 242–246 (2003).
26. Q.-H. Ling, M. Q.-H. Meng, and T. Mei, "Robotic Pet Based Interactive Home Healthcare System," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, CA (2005), pp. 771–776.
27. C. Paggetti and E. Tamburini, "Homecare Service Perspective in Tuscany: Vision and New User Centered Services," *Studies in Health Technology and Informatics*, 137–146 (2005).
28. A. Bar-Or, D. Goddeau, J. Healey, L. Kontothanassis, B. Logan, A. Nelson, and JM Van Thong, "BioStream: A System Architecture for Real-Time Processing of Physiological Signals," *Proceedings of the 26th Annual International Conference of IEEE Engineering in Medicine and Biology Society*, San Francisco, CA (2004), pp. 3101–3104.
29. K. Lorincz, D. J. Malan, T. R. F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Manland, M. Welsh, and S. Moulton, "Sensor Networks for Emergency Response: Challenges and Opportunities," *IEEE Pervasive Computing* **3**, No. 4, 16–23 (2004).
30. Wireless Sensor Networks, Crossbow Technology Inc., http://www.xbow.com/Products/Wireless_Sensor_Networks.htm.
31. G. Amato, S. Chessa, F. Conforti, A. Macerata, and C. Marchesi, "Networking and Data Management for Health Care Monitoring of Mobile Patients," in *Personalized Health Management Systems: The Integration of Innovation Sensing, Textile, Information and Communication Technologies*, C. D. Nugent, P. J. McCullagh, E. T. McAdams, and A. Lymberis, Editors, IOS Press, Fairfax, VA (2005), pp. 191–200.
32. Experts Discuss Progress of NHIN Prototypes, S. Weier, Editor, *iHealthBeat*, <http://www.ihealthbeat.org/index.cfm?action=dspItem&itemID=118988&changedID=118969>.
33. CEN ENV13734, "Health Informatics—Vital Signs Information Representation," Technical Committee for Medical Informatics (TC251) of the European Committee for Standardization (CEN) CEN/TC251/PT5-021, Final Draft (July 1999).
34. M. Alwan, S. Dalal, D. Mack, S. Kell, B. Turner, J. Leachtenauer, and R. Felder, "Impact of Monitoring Technology in Assisted Living: Outcome Pilot," *IEEE Transactions on Information Technology in Biomedicine* **10**, No. 1, 192–198 (2006).
35. J. A. Stankovic, I. Lee, A. Mok, and R. Rajkumar, "Opportunities and Obligations for Physical Computing Systems," *Computer* **38**, No. 11, 23–31 (2005).
36. A. G. Mathews and R. Butler, "A Vision for the Use of Proactive Mobile Computing Tools to Empower People with Chronic Conditions," *Proceedings of 18th IEEE Symposium on Computer-Based Medical Systems*, Dublin, Ireland (2005), pp. 425–427.

Accepted for publication August 21, 2006.

Published online December 19, 2006.

Marion Blount

IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (mlblount@us.ibm.com). Dr. Blount is a research staff member in the Next Generation Web department. He received a B.S. degree in electrical engineering from Duke University and M.S. and Ph.D. degrees in electrical engineering from Stanford University. He was a participant in the Pervasive Care Connect (PCC) design and development teams. Dr. Blount's most recent research projects have been in the areas of pervasive computing and health care.

Virinder M. Batra

IBM Software Group, 4205 South Miami Boulevard, Durham, North Carolina 27703 (batra@us.ibm.com). Mr. Batra is a senior software engineer manager in IBM Healthcare and Life Sciences. He received a B.S. degree in electrical engineering from the Birla Institute of Technology and Sciences, India, an M.S. degree in computer sciences from Virginia Polytechnic Institute, and an M.B.A. degree from the University of North Carolina. He has been a product development lead for a variety of IBM products including NetView® and the WebSphere Application Server. Mr. Batra recently managed the development of the PCC and the IBM Solution for Compliance in a Regulated Environment (SCORE).

Andrew N. Capella

IBM Software Group, 4205 South Miami Boulevard, Durham, North Carolina 27703 (capella@us.ibm.com). Mr. Capella is a senior software engineer in IBM Healthcare and Life Sciences.

He received a B.S. degree in electrical engineering from Ohio State University and an M.S. degree in computer engineering from Syracuse University. Mr. Capella has led various development teams, including TCP/IP, WebSphere Application Server, Location Based Services, and Pervasive Care and Case Management.

Maria R. Ebling

IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (ebbling@us.ibm.com). Dr. Ebling is a research staff member and manager of the Privacy-Enabled Context Technologies department. She received a B.S. degree in mathematics with a computer science option from Harvey Mudd College, and M.S. and Ph.D. degrees in computer science from Carnegie Mellon University. She has worked in the area of context-aware computing, most recently in applying context-aware computing in the area of health care. She received an IBM Technical Group Award for her work on intelligent notification. Dr. Ebling is a member of the IEEE Computer Society and the Association for Computing Machinery.

William F. Jerome

IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (wfj@us.ibm.com). Mr. Jerome is a senior software engineer in the Privacy-Enabled Context Technologies department. He received a B.S. degree in computer science from Brooklyn College and an M.S. degree in computer science from Columbia University. His recent research projects have been in presence awareness, health care, and intelligent notification. Mr. Jerome is a member of the Association for Computing Machinery.

Sherri M. Martin

IBM Software Group, 11501 Burnet Road, Austin, Texas 78758 (sherrim@us.ibm.com). Ms. Martin is a senior software engineer in Healthcare and Life Sciences Solutions Development, working on pervasive solutions in health care. She received a B.S. degree in computer information systems from the University of Florida and an M.B.A. from Florida Atlantic University. She has worked on and managed a number of development projects, including the IBM SecureWay® Directory and OS/2®. She was the development-offering lead for PCC. Ms. Martin is a member of the Leadership Management Institute at the University of Texas.

Michael Nidd

IBM Research GmbH, Zurich Research Laboratory, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland (mni@zurich.ibm.com). Dr. Nidd is currently studying the use of short-range wireless networks in health data collection. Prior work included what is now called the “converged telephone,” data transmission formats over digital audio broadcasting, and signal strength prediction for indoor wireless networks. He earned his Bachelor’s and Master’s degrees in mathematics from the University of Waterloo, Canada, and his Docteur ès Sciences degree from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

Michael R. Niemi

IBM Software Group, 8051 Congress Avenue, Boca Raton, Florida 33487 (niemi@us.ibm.com). Mr. Niemi is a senior software engineer working on IBM Pervasive Solutions and support for standards in Healthcare and Life Sciences. He received a B.S. degree in computer science from Rutgers University. Mr. Niemi has worked on a variety of hardware and software projects including logic design and performance analysis on IBM mainframes, TCP/IP, WebSphere Application Server, WebSphere Voice Server, Pervasive Solutions, and standards in Healthcare and Life Sciences.

Steven P. (Steve) Wright

IBM United Kingdom Limited, Hursley Park, Winchester, Hampshire, England SO21 2JN (steve_wright@uk.ibm.com). Mr. Wright is a Certified Consulting IT Specialist. Since joining IBM he has worked in various development organizations including Video Conferencing, Voice Technology, and WebSphere MQ. He has wide experience of customer engagements and, more recently in the Pervasive Group, he has worked on the Pervasive Care and Case Management project. ■